ARIFICIAL INTELLIGENCE AND TUTORIAL SYSTEMS

HOW IT WORKS

The program is all task lead.
This means that we maintain a list of tasks, when we sort through to decide what happens.
New tasks are set up from game messages, which occur from events in the game, or from user input.

OVERVIEW OF DATA STRUCTURES

MESSAGE LIST
;game turn events, fresh every game turn cycle. These provide triggers for tasks and events

TASK LIST
;actions to be performed / assigned. These are active until they are completed.
AI Routines
;code which sorts through the Messages and sets up new tasks

EVENTS
;things that directly happen;
;usually carried out immediately the next processed turn. E.G. play a sound effect
;some are continuous, like move unit until it reached position X,Y,Z

UNIT LISTS
;We will have lists of active units, which will contain many individual variables. The Variables relevant to this section are unit type, X,Y,Z position, task assigned

AI ROUTINES

The AI is specific code written to process the messages, tasks and events.

TASK LIST DATA

The format of the Task List data.

Task No	Started flag	Assigned Flag	Item Ref no.	Waiting count	Priority	Assignment	Time IN /OUT	Task Type	Pos

TASK NO.

This unique reference, allows us to reference specific tasks together. This allows us to track events, to break or reassign if needed.
E.G A man is walking to a position to collect a crystal. The task is assigned to him. If he gets zapped by a rock monster on the way, the unique link, via the task number, will allow that task to be 'freed up' ready to be reassigned.
We can have a counter which is incremented each time it is used, this will make it unique until it wraps around.

STARTED FLAG

This shows a task has started but not finished, so it is not ready to be removed from the task list yet. A started task is less likely to be given to another unit that a started task. It also allows us to discover what work that is currently being done. This may be used by the AI to automatically share the work load out.
E.G say, digging has twice the priority of collecting crystals, if we have 10 units currently digging and only 2 collecting crystals, then we will want to assign units to collect crystals not digging, although it has twice the priority.

ASSIGNED FLAG

This flag is show that the task has been assigned. But not yet started.
E.G. The task is digging, A man is on his way but not there digging yet.
(This needs to be used carefully, as it may tie up a task, which will prevent others from starting or assisting)

ITEM REF NO.

Item Reference number, this is a link to reference a specific unit which could be either a vehicle building, lego man, rock monster etc.

WAITING COUNT

This increases as the task is left in the list. We want to increment this counter every game turn. It lets us find out how long a task has been waiting to be assigned. A task that is left for some time may increase in priority, or may be removed, we will test to find out what is playable.

PRIORITY

This is the current level of priority given to this task. Each task is assigned a default priority, the list of priorities can be altered by the player during the game. It is one of the main priorities for sorting tasks.
Priorities set from 0 to 100.
Normal priority 50.

Player assigned tasks will always be higher than the normal priority for the same task

This will be tweakable but initially add 10, they may all be 90+?

Priority 100 is must do. Remove unit from other tasks and reassign this task.

ASSIGNMENT

Was this task assigned by a player or auto assigned by the AI.
Player tasks will be treated differently. They may initially be low priority, but if left they will increase in priority (The player must be obeyed)

TIMEOUT

Some tasks must be performed within a certain time, these will automatically be removed from the list when this counter gets set down to 0. The counter is decremented by 1 every game loop
E.G sets a trap for an earthquake if any unit walks here, disappears after 2 minutes

TIMEIN

Some tasks are set and will appear after a certain time, this has a counter which counts down to 0 and then gets a high priority.
E.G a reminder in 'one minute' that the base is under attack'

AI finds the player is doing well, set up a rock monster to arrive in 10 seconds

TASK TYPE

There are different task types, these allow us to sort the tasks more efficiently. We will need to assign a number for each different task types
E.G. 1=move to

2=dig

36,589=build a teleporter.

For speed of checking these can be grouped e.g. over 1 to 999 are men events 1000-1999 are buldings etc.

POSITION X,Y,Z

The map position that this event wants to occur at. This will be block position's unless the task needs to be more specific.

GAME LOOP

PROCESS TASK LIST

The task list can be thought of as a single list or at times split into various sections of smaller lists.
At times only relevant sections of the list are needed, and it may actually be better to sort it into different lists.
We may split it into UnAssigned, assigned and started tasks, as they are processed separately.

For all the UnAssigned Tasks
Increase 'waiting count' of task
Decrease 'timeout' of task
    If 0 remove task
Decrease 'Timein' of task
    If 0 set priority to max.

Process all the non processed tasks first this will allow started units to be reassigned, and non started tasks to be started and then processed the same turn.

TASK NOT ASSIGNED

The exact order of sorting will depend on options and difficulty levels that the player will select.
Initially we will sort the unassigned tasks into priority the highest priority tasks being allocated first.

If a task is not assigned – try and assign it. This will involve a number of tests and checks which depend on each individual task, these will be documented later.

TASK ASSIGNED

If a task has be assigned – currently no checks for this.

TASK STARTED

If a task has started – check if it has finished. This will involve a number of tests and checks which depend on each individual task, these will be documented later.

TASK COMPLETE

If a task has finished, Remove it from the list and set up the relevant PTL data for that task..
E.G dig block task, sets off 'rock fall' & random avalanche tasks:
    rock fall then sets off pick up debris task.
If a unit has finished the task, set off 'check block' message

Later optimisation
If an 'assigned' unit is in the same X,Y,Z position as a task that has not yet been 'Assigned', and it can do the task and the new tasks priority is higher, then reassign the unit to the new task.

Multiple units may perform the same task. E.G. Two men digging the same rock section out.

TESTS TO ASSIGN UNITS A TASK

What Tests a unit needs to pass to allow the AI to assign the task to that unit.
Initially we can assign the first unit that can do a task to start doing that task.
We can then program a more 'intelligent' check to find a better match (this may be part of the difficulty settings)

A better match would depend on –

- The nearest unit to the position
- The unit able to get to the position fastest
- The strongest unit (this could be the fastest digger, or best fighter)
- It could take into account the units current health / damage
- Other tasks in the list which it may be better suited to do.
- The players choice of priorities

General	Is the unit alive Is the unit not currently doing a started task
Dig here	Is the unit able to dig Can it reach the position.

GAME MESSAGES

The game code generates a number of messages which tells the AI what is going on. These are stored in a Message list as the turn is processed,
Later the AI filters through the list to decide which tasks it will set up.

Examples of game messages are:

- Unit X under attack
- Building under attack
- Save button clicked.
- Dig selected on wall X,Y,Z
- Men type X selected.

By using the game messages to set up tasks then we can, automate many things in the game by setting off a message. This will be useful for demonstrating to the player causes and effects in the tutorials. It will also reduce the data for networking by sending only message data rather than larger event data.

PTL

.ptl files. this stands for PROCESS TASK LISTS
Lists of Tasks or Events which are to be processed when a event occurs

There is a PTL list for every event,
It is split into two separate lists one for starting the event and another list for when the task has completed.
These lists can be copied straight into the message or task lists.
They will be constructed from a user friendly editor which we will write.

EG. If selecting a 'man type X' was game event 23
The PTL file for event 23 would contain

For task started
1. play a sfx
2. change the icon bar to show that lego mans icon options
3. print a selected bar around the man
4. centre the screen at x,y,z

and nothing for task completed

When the unit was selected all these tasks would be able to be copied to the TASK list.
We may later decide to add, other additional events
E.G 'clear any other clicks from the AI list' (because the icon menu has changed)
And another would be 'deselect all other units'
We could edit the task 23 PTL file and add these tasks

For task started
1. deselect all units
2. play a sfx
3. change the icon bar to show that lego mans icon options
4. print a selected bar around the man
5. centre the screen at x,y,z
6. clear any other clicks from the AI list

and for task completed
1. deselect all units
2. remove icon bar
3. restore default icon bar

As you can hopefully see, the game would be driven from these tasks, they would be flexible and easy to change.

Some tasks are only performed when at a certain X,Y,Z position,
This allows us to improve the search by excluding some tasks in the list as units not at that position, do not need to process/check.

Some options we will not deal with now but we have the option to extend later if and as when needed. The exact usage will be determined by the game play.

E.G.
An assigned task, does not mean that that task is complete, it is still in the task list. If a unit is ‘waiting’ we may assign it to also do an ‘assigned task’ as extra help. This may be optional with high priority cases sent more units, to make sure the work is done!

EXAMPLE GAME TURN

Below are two examples of what happens to the various tables and lists in the game.
It shows two game actions:
1.The player clicks on a unit.
2.The player selects a block to dig.

INPUT ACTION	MESSAGE LIST	AI processing code	TASK LIST	EVENT
1.Click on a Lego man	Unit Selected 'n'	Found 'new unit selected' in messages, load .PTL file Set up all the tasks and events in this list.		
				1. play a sfx 2. change the icon bar to show that lego mans icon options 3. print a selected bar around the man 4. centre the screen at x,y,z
2.Dig at this spot	Dig X,Y,Z			
		Read 'dig' game message set up task in list	'xxx' DIG X,Y,Z	
		1. Found an order in the AI list 'DIG X,Y,Z' 2. Searched the list of units available and found a lego man that can dig and is currently waiting. 3. Assigned him the event 'Move to X,Y,Z' linked with 'xxx' task set as started.	'xxx' DIG X,Y,Z 'xxx' Move to X,Y,Z	

		4. Flagged the 'DIG' task as assigned & linked with the unique number.		
		1. No messages 2. Check task list 3. 'Move to', has started so skip 4. 'Dig' is assigned so skip This repeats for each game turn while the man walks!	'xxx' DIG X,Y,Z 'xxx' Move to X,Y,Z	
		1. Man walking task finished 2. Remove 'move to' task 3. Set up message – check block X,Y,Z	'xxx' DIG X,Y,Z	
	Check Block X,Y,Z	1. Found 'check block' message, 2. Search task list for task at this position 3. Found 'DIG X,Y,Z' 4. Assign unique number 'xxx' dig task to man 5. Set 'Dig' task to started.	'xxx' DIG X,Y,Z	
		The game continues until the digging is completed. Then the digging complete is signalled and the PTL data for completed task is inserted,	'xxx' DIG X,Y,Z	
		1. Dig task completed 2. Remove 'dig' task 3. Insert dig-PTL data 4. Set up message – check block X,Y,Z	ROCK SLIDE X,Y,Z	

	Check Block X,Y,Z	1. Found 'check block' message, 2. Search task list for task at this position - None found 3. Rock Slide continues	ROCK SLIDE X,Y,Z	
		Rock Slide continues until complete then its Completed PTL data is added	Collect Crystal at X,Y,Z Clear Rubble at X,Y,Z	
		1. AI searches task list and finds 'Collect crystal' and 'Clear rubble' tasks 2. Clear Rubble' task has a higher priority (the player chose it! 3. it searches the unit list and finds a man in the same spot, it assigns him the clear rubble task 4. It then tries to assign Collect Crystal' task but there is no units available in the list.	Collect Crystal at X,Y,Z Clear Rubble at X,Y,Z	
		1. The game continues unit the rubble is cleared. 2. The 'Clear Rubble' task is removed, there is no PTL data. 3. Set up message – check block X,Y,Z	Clear Rubble at X,Y,Z Collect Crystal at X,Y,Z	
	check block X,Y,Z	1. Found 'check block' message, 2. Search task list for task at this position -	Collect Crystal at X,Y,Z	

		3. The 'Collect Crystal' message is found and the unit is assigned to this task		
		The collect crystal task is completed and the PTL data is added	Crystals to nearest REFINERY	
		The unit goes to the Refinery		

PTL DATA for the above example

DIG COMPLETED ROCK SLIDE X,Y,Z
ROCK SLIDE COMPLETED Clear Rubble at X,Y,Z Collect Crystal at X,Y,Z
UNIT SELECTED play a sfx change the icon bar to show that lego mans icon options print a selected bar around the man centre the screen at x,y,z
CLEAR RUBBLE completed
COLLECT CRYSTAL Completed Crystals to nearest REFINERY

SELECTED UNITS

If a task is about to be entered into the task list and a unit is currently selected then check if that unit can be assigned the task, first before putting it in the task list.
If a group is selected then check if each unit is able to perform the task and assign them all to the task.

TASKS

Move To X,Y,Z	Move a unit to..
Dig Small tunnel X,Y,Z	
Dig big tunnel X,Y,Z	
Collect Crystal X,Y,Z	
Collect Ore X,Y,Z	
Collect Rubble X,Y,Z	
Build Support X,Y,Z	
Defend spot X,Y,Z	
Gather men X,Y,Z	
Place dynamite X,Y,Z	
Build barricade X,Y,Z	
Build Building n X,Y,Z	For each building type E.g. Teleport
Build man n X,Y,Z	For each Lego model E,G geologist
Build vehicle n X,Y,Z	For each Lego vehicle e.g. digger
Teleport ore	
Switch power On/Off	
Crystals to nearest REFINERY	

GAME MESSAGES

Check Block X,Y,Z	When one task is finished, get the unit to check the block for other tasks before re-assigning.
Selected unit n	Unit n has been selected
Selected group X,Y,Z	Selected a group of units, centered at X,Y,Z
Icons selected	Set up for each and every icon as selected.
Unit n under attack	Unit n is being attacked
Building under attack	
Dig selected on wall X,Y,Z	
Crystal found at X,Y,Z	Set this up for each crystal located
Ore found at X,Y,Z	Set this up for each ore rock located

EVENTS

Play SFX	
Display Icon panel	

TUTORIALS

The tutorials are split into three sections which the player can choose any to start with, and at the end he is offered the chance of progressing onto the next turial or starting the game. The player will be allowed to skip out of the tutorial at and time.

Control options tutorial

These are designed to demonstrate each basic control function of the game. It is an Ideal introduction for a younger player, who has not played this type of game before, and will be unfamiliar with the options, control and general design and effect.

- All of the messages are played as sampled speech, this will allow the player to easily understand the tutorial objective, and perform the actions needed, without slowing down to read text.
- All of the spoken text is duplicated as scrolling text in the message window at the bottom of the screen, so users without a sound card can still understand the tutorial and it will serve as a reminder and strengthen the message for the other players.
- Where an object, is mentioned it will also display a graphic or icon to show what the object is.
- This will allow the maximum understanding of the tutorial information.
- Audiably the message is spoken
- Visually the text is displayed.
- The information is not complicated with jargon, as it graphically displays all objects for visual recognition, and strengthens the name/ graphic bond.
- The tutorials are lead through with characters to individualise the demonstration and encourage the understanding through identification with the named characters.
- Many options are deliberately not selectable, to prevent accidental selection, until that aspect of the game options has been explained.
- The tutorials will expand on the ideas, by allowing the player to combine actions, these will result in rewards to encourage the player.
- The tutorials are linked with the narrative and characters following a progressive story-line, which introduces the different options.

SELECTION

Introduce the cursor
For this tutorial the level map is only slightly larger than one screen wide, This will prevent the player scrolling away from the action but they may ‘accidentally’ notice the cause and effect of scrolling. (limit but do not hinder the players learning ability)

Show a cave, with Roxy in the middle.
CAPT: “This is ROXY the pilot of the Rock Raiders team”.
ANIM of Roxy waving to the screen
ROXY: “hello”

CAPT: “We are going to help Roxy (pic), gather some power crystals (pic). They are needed to power our spaceship”

CAPT: "If you move your mouse around you will see the cursor (pic) moving."
"move it over the top of ROXY (pic)
ANIM Roxy waves again.
ROXY: "hello"

a graphic of an object and a short description, ask the player to try and click on the same object on the level. This will be for -

1. A Lego man,
2. then a Power crystal
3. and a Teleport building.

SCROLLING, ZOOMING AND ICONS

Now we explain about the 'world' is larger than just the one screen,

1. You can view the level by scrolling the map. This is done by moving the cursor to the side of the screen
2. If you want to see more of the level at one time then you can zoom out. use the zoom out icon to see the whole map
3. and now use the zoom in to return to the initial screen
4. There are a number of other icons that make things happen in the game, the MAP icon at top right opens a window to show us another view of the whole level shruk down into a very small scale.
5. If you scroll the level again, by moving the cursor to the edges of the screen, Notice how the map display changes to show where you are in the level.
6. Click on the MAP icon again to make the map disappear.

MOVING

We can use these new skills you have learnt to start playing the game.

1. First select a lego man by clicking on him (option to re-try the selection tutorial)

Now we need to tell him to do something.
One the right side of the screen (point arrow) you will have noticed that we have lots of new icons which we can choose from. All of these make

MOVEMENT
COLLECTION

BASIC TUTORIAL

ADVANCED TUTORIAL